SAML V2.0 Holder-of-Key Web Browser SSO Profile

Working Draft 067

26 August 22 September 2008

Specification URIs:

This Version:
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-draft-07.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-draft-07.odt
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-draft-07.pdf

Previous Version:
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-draft-06.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-draft-06.odt
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-draft-06.pdf

Latest Version:
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.odt
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.pdf

Technical Committee:
OASIS Security Services TC

Chair(s):
Hal Lockhart, BEA Systems, Inc.
Brian Campbell, Ping Identity Corporation

Editor(s):
Nate Klingenstein, Internet2

Related Work:
This specification is an alternative to the SAML V2.0 Web Browser SSO Profile in the SAML V2.0 Profiles specification [SAML2Prof].

Declared XML Namespace(s):
urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key;SSO:browser:holder-of-key

Abstract:
This profile allows for transport and validation of holder-of-key assertions by standard HTTP user agents with no modification of client software and maximum compatibility with existing
deployments. Most of the flows are as in standard Web Browser SSO, but an X.509 certificate presented by the user agent supplies a valid keypair through client TLS authentication for HTTP transactions. Proof of key possession resulting from TLS authentication is used for holder-of-key validation of a SAML assertion. This strengthens the assurance of the resulting authentication context and protects against credential theft, giving the service provider fresh authentication and attribute information without requiring it to perform successful validation of the certificate.

**Status:**

This document was last revised or approved by the SSTC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at [http://www.oasis-open.org/committees/security](http://www.oasis-open.org/committees/security).

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page ([http://www.oasis-open.org/committees/security/ipr.php](http://www.oasis-open.org/committees/security/ipr.php)).

The non-normative errata page for this specification is located at [http://www.oasis-open.org/committees/security](http://www.oasis-open.org/committees/security).
Notices

Copyright © OASIS® 2008. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.
1 Introduction

In the scenario addressed by this profile, which is an extended version of the Web Browser SSO Profile in section 4.1 of [SAML2Prof], a principal uses an HTTP user agent to access a web-based resource at a service provider. To do so, the user agent needs to acquire a SAML assertion from its preferred identity provider. The user may first acquire a request for authentication from the service provider or a third party. The user agent then makes a request to the identity provider using client TLS authentication. The X.509 certificate supplied in this transaction is used primarily to supply a public key that is associated with the principal. The identity provider authenticates the principal by way of this TLS authentication or any other method of its choosing. The identity provider then produces a response containing at least an assertion with holder-of-key subject confirmation and an authentication statement for the user agent to transport to the service provider. This assertion is presented by the user agent to the service provider over client TLS authentication to prove possession of the private key matching the holder-of-key confirmation in the assertion. The service provider should rely on no information from the certificate beyond the key to process the assertion. The assertion is consumed to create a security context. The TLS key may then be used to persist the security context rather than a cookie or other application-layer session.

To implement this scenario, a profile of the SAML Authentication Request protocol is used in conjunction with the HTTP Redirect, HTTP POST and HTTP Artifact bindings. It is assumed that the user is using an HTTP user agent capable of presenting client certificates during TLS session establishment, such as a standard web browser.

1.1 Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC 2119].

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

Conventional XML namespace prefixes are used throughout this specification to stand for their respective namespaces as follows:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>XML Namespace</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>md:</td>
<td>urn:oasis:names:tc:SAML:2.0:metadata</td>
<td>This is the SAML V2.0 metadata namespace defined in the SAML V2.0 metadata specification [SAML2Meta].</td>
</tr>
<tr>
<td>ds:</td>
<td><a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a></td>
<td>This is the XML digital signature namespace defined in the XML Signature Syntax and Processing specification [DSig].</td>
</tr>
<tr>
<td>hok:</td>
<td>urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser:holder-of-key</td>
<td>This is the web browser holder-of-key namespace is defined by this document and its accompanying schema [HoK-XSD].</td>
</tr>
<tr>
<td>saml:</td>
<td>urn:oasis:names:tc:SAML:2.0:assertion</td>
<td>This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification [SAML2Core].</td>
</tr>
<tr>
<td>samlp:</td>
<td>urn:oasis:names:tc:SAML:2.0:protocol</td>
<td>This is the SAML V2.0 protocol namespace defined in the SAML V2.0 core specification [SAML2Core].</td>
</tr>
</tbody>
</table>
This specification uses the following typographical conventions in text: <namespace:Element>, Attribute, Datatype, OtherKeyword.

For purposes of this document, Transport Layer Security (TLS) as defined in [RFC 4346] includes Secure Sockets Layer V3.0.

1.2 Normative References


1.3 Non-normative References


1.4 Conformance

1.4.1 Identity Provider

A conforming implementation of an identity provider MUST support the following normative text of this profile: sections 2.4.3, 2.4.4, 2.4.5, 2.5.2, and 2.5.3. If the identity provider uses metadata, it MUST also support section 2.6.

In addition to the typical requirements for a SAML implementation, a conforming identity provider MUST meet the conformance requirements listed in [SAML2HoKAP]. Support the following functionality to support interoperability:

- Retrieving a certificate presented by a user agent and validating that the user agent possesses the corresponding private key through TLS; and
- Encoding this certificate using Base64 as <ds:KeyInfo> in a <saml:SubjectConfirmationData> element.

1.4.2 Service Provider

A conforming implementation of a service provider MUST support the following normative text of this profile: sections 2.4.1, 2.4.2, 2.4.3, 2.4.6, 2.5.1, and 2.5.4. If the service provider uses metadata, it MUST also support section 2.6.

In addition to the typical requirements for a SAML implementation, a conforming service provider MUST meet the conformance requirements listed in [SAML2HoKAP]. Support the following functionality to support interoperability:

- Retrieving a certificate presented by a user agent and validating that the user agent possesses the corresponding private key through TLS; and
- Decoding a Base64-encoded certificate within a <ds:KeyInfo> in a <saml:SubjectConfirmationData> element and matching it against a presented certificate.
2 Holder-of-Key Web Browser SSO Profile

2.1 Required Information

**Identification:** urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser:holder-of-key

**Contact information:** security-services-comment@lists.oasis-open.org

**SAML Confirmation Method Identifiers:** The SAML V2.0 “holder-of-key” confirmation method identifier, urn:oasis:names:tc:SAML:2.0:cm:holder-of-key, is included in all assertions issued under this profile.

**Description:** Given below.

**Updates:** Provides an alternative to the SAML V2.0 Web Browser SSO Profile given in section 4.1 of [SAML2Prof].

2.2 Background

This profile is designed to enhance the security of SAML assertion and message exchange without requiring modifications to client software. The SAML assertion is delivered to the service provider over mutually authenticated TLS using keying information vetted by the identity provider, resulting in strong association of the resulting security context with the intended user and elimination of many attacks. Additional usability benefits can be realized by leveraging the end user's certificate for discovery and authentication.

Deployments should minimize user interaction, avoid mutually conflicting CA requirements, and must ensure presentation of the same certificate to all services by coordinating certificate issuance and TLS configuration.

If both the identity provider and service provider use this profile, but assume no knowledge of the certificate's contents, enhanced security is the primary benefit. There is a small chance that a bearer token will be stolen in transit, as described in [SAML2Secure]. Confirming that the presenter of the token is the intended holder through public key cryptography virtually eliminates this chance, improving the viability of SAML-based HTTP SSO for sensitive applications.

If a certificate can be used by the identity provider for principal authentication, there is no need for the user to further confirm its identity, and potentially no user interaction is needed. Further, if the user accesses the service provider first, discovery of the user's identity provider may be performed by matching fields within the certificate presented; however, that is beyond the scope of this specification.

This profile offers meaningful advantages over traditional PKI, as well. There is no requirement for a mutually or universally trusted root, distributed OCSP or CRL-based revocation, a globally unique namespace, PKI validation (particularly by the SP), or for all participants in SSO to utilize X.509. The authentication token can be customized for every transaction, including fresh attributes and appropriate revelation of identity.

2.3 Profile Overview

Figure 1 illustrates the basic template for achieving SSO. The following steps are described by the profile. Within an individual step, there may be one or more actual message exchanges depending on the binding used for that step and other implementation-dependent behavior.
1. **HTTP Request to Service Provider**
   The principal, via an HTTP user agent, makes an HTTP request for a secured resource at the service provider. This request may or may not be made over TLS. The service provider determines that no security context exists, and attempts to create one.

2. **Service Provider Determines Identity Provider**
   The service provider determines the appropriate identity provider to which to direct the user agent. This may be done through use of a discovery service as described in [IDPDisco], by examining fields in a certificate presented through client TLS authentication, such the X.509 subject or subjectAltName, or by any other means appropriate.

3. **<samlp:AuthnRequest> issued by Service Provider to Identity Provider**
   The service provider issues a <samlp:AuthnRequest> message to be delivered by the user agent to the identity provider. The HTTP Redirect, HTTP POST, or HTTP Artifact binding can be used to transport the message to the identity provider through the user agent. The user agent presents this message in a request to the identity provider using TLS.

4. **Identity Provider identifies Principal**
   The principal is identified by the identity provider. The identity provider identifies the principal using any authentication method at its discretion honoring any requirements imposed by the service provider in the
<samlp:AuthnRequest>, including validation of the certificate presented in client TLS authentication. However, the identity provider must establish that the private key corresponding to the X.509 certificate that will be included for holder-of-key proofing is held by this user agent.

5. Identity Provider issues <samlp:Response> to Service Provider

The identity provider issues a <samlp:Response> message to be delivered by the user agent to the service provider. The user agent presents this response to the service provider using TLS. Either the HTTP POST or HTTP Artifact binding can be used to transfer the message to the service provider through the user agent. The message may indicate an error or will include at least an authentication statement in an assertion with holder-of-key <saml:SubjectConfirmation> containing an X.509 certificate associated with the principal.

6. Service Provider grants or denies access to Principal

The response is received by the service provider, which can respond to the principal's user agent by either establishing a security context for the principal and returning the requested resource or returning its own error or an error passed by the identity provider.

Note that an identity provider can initiate this profile at step 5 by issuing a <samlp:Response> message to a service provider without the preceding steps. The user agent or a third party may also initiate this profile by spoofing the authentication request if there is no requirement it be signed.

2.4 Profile Description

If the profile is initiated by the service provider, start with section 2.4.1. If the request is unsigned and spoofed by the user agent or a third party, start with section 2.4.4. If initiated by the identity provider, start with section 2.4.5. The descriptions refer to a single sign-on service Single Sign-On Service and Assertion Consumer Service assertion consumer service in accordance with their use in section 4.1.3 of [SAML2Prof]. Processing rules for all messages are specified in section 2.5.

2.4.1 HTTP Request to Service Provider

The profile may be initiated by an arbitrary HTTP request to the service provider. The service provider is free to use any means it wishes to associate the subsequent interactions with the original request. Each of the bindings provides a RelayState mechanism that the service provider MAY use to associate the profile exchange with the original request. If a TLS session is established for this initial request, it MAY be used for discovery in section 2.4.2.

2.4.2 Service Provider Determines Identity Provider

The service provider determines the identity provider with which the principal is associated through a variety of mechanisms as selected by the service provider implementation or deployment. If the initial request in section 2.4.1 was made over mutually authenticated TLS, the service provider MAY check the certificate presented by the user agent and use the X.509 subject, subjectAltName, or other field or extension in the certificate to determine the principal's identity provider or single sign-on service endpoint.

2.4.3 <samlp:AuthnRequest> Issued by Service Provider to Identity Provider

Once an identity provider is selected, the location of a single sign-on service to which to send a <samlp:AuthnRequest> is determined based on the SAML binding chosen by the service provider. Metadata as described in section 2.6 MAY be used for this purpose. Following an HTTP request by the
user agent, an HTTP response is returned containing a `<samlp:AuthnRequest>` message or an artifact, depending on the SAML binding used, to be delivered to the identity provider's single sign-on service.

Profile-specific rules for the contents of the `<samlp:AuthnRequest>` are defined in section 2.5.1. The `<samlp:AuthnRequest>` message should be signed if the identity provider requires the request issuer to be verified and the HTTP Redirect or HTTP POST bindings are used; if the HTTP Artifact binding is used, then the request issuer can be verified through other means.

### 2.4.4 Identity Provider Identifies Principal and Verifies Key Possession

If the HTTP Redirect or POST binding is used, a `<samlp:AuthnRequest>` message is delivered directly to the identity provider to begin this step. If the HTTP Artifact binding is used, the Artifact Resolution profile defined in section 5 of [SAML2Prof] is used by the identity provider, which makes a callback to the service provider to retrieve the `<samlp:AuthnRequest>` message using, for example, the SOAP binding.

The identity provider must perform two functions in this step: identification of the principal presenting the `<samlp:AuthnRequest>`, and verification that the principal possesses the private key associated with the X.509 certificate that will be included as `<saml:SubjectConfirmation>`.

The identity provider MUST establish the identity of the principal (unless it will return an error) prior to the issuance of the `<samlp:Response>`. If the `<samlp:AuthnRequest>` attribute `ForceAuthn` is present and true, the identity provider MUST freshly establish this identity rather than relying on any existing session it may have with the principal. Otherwise, and in all other respects, the identity provider may use any means to authenticate the user agent, subject to any requirements included in the `<samlp:AuthnRequest>`.

It is REQUIRED that the `<samlp:AuthnRequest>` be presented to the identity provider over mutually authenticated TLS to supply the identity provider with the X.509 certificate and establish the user agent's possession of the corresponding private key. The certificate resulting from this process MUST match the one included as holder-of-key `<saml:SubjectConfirmation>` in the subsequent `<samlp:Response>`.

### 2.4.5 Identity Provider Issues `<samlp:Response>`

The identity provider presents an HTTP response to the user agent containing a `<samlp:Response>` message or an artifact, depending on the SAML binding used, to be delivered to the service provider's assertion consumer service. The exact format of this HTTP response and the subsequent HTTP request to the assertion consumer service is defined by [SAML2Bind].

- If the HTTP POST binding is used, the `<samlp:Response>` message is delivered directly to the service provider in this step.
- The HTTP Redirect binding MUST NOT be used, as the response will typically exceed the URL length permitted by most user agents.

Profile-specific rules on the contents of the `<samlp:Response>` are included in section 2.5.3.

The location of the assertion consumer service MAY be determined using metadata as described in section 2.6. The identity provider MUST have some means to establish that this location is in fact controlled by the service provider. A service provider MAY indicate the SAML binding and the specific assertion consumer service to use in its `<samlp:AuthnRequest>` and the identity provider MUST honor them if it can.
2.4.6 Service Provider Grants or Denies Access to Principal

The HTTP request presenting the message resulting from section 2.4.5 to the service provider MUST be made over mutually authenticated TLS to demonstrate possession of the private key corresponding to the certificate included in the assertion’s <saml:SubjectConfirmation> as well as maintain confidentiality and message integrity. The <saml:Assertion> element(s) in the <samlp:Response> MUST be signed if the HTTP POST binding is used, and MAY be signed if the HTTP Artifact binding is used.

If the HTTP Artifact binding is used, the Artifact Resolution profile defined in section 5 of [SAML2Prof] is used by the service provider, which makes a callback to the identity provider to retrieve the <samlp:Response> message, using for example the SOAP binding. The front-channel TLS session could be used to persist client state during artifact resolution, or establish state afterwards by claiming a resolved assertion.

To complete the profile, the service provider processes the <samlp:Response> and <saml:Assertion>(s) and creates a security context for the user. The service provider MUST process the <samlp:Response> message and any enclosed <saml:Assertion> elements as described in [SAML2Core].

The service provider MAY establish a security context with the user agent using any session mechanism it chooses. Any subsequent use of the <saml:Assertion>(s) provided is at the discretion of the service provider and other relying parties, subject to any restrictions on use contained within them.

2.5 Use of Authentication Request Protocol

This profile uses the Authentication Request protocol defined in [SAML2Core]. In the nomenclature of actors enumerated in section 3.4 of that document, the service provider is the request issuer and the relying party, the user agent is the attesting entity and presenter, and the principal is the requested subject. There may be additional relying parties at the discretion of the identity provider.

2.5.1 <samlp:AuthnRequest> Usage

A service provider MAY include any <samlp:AuthnRequest> message content described in [SAML2Core], section 3.4.1. All processing rules are as defined in [SAML2Core]. The request MUST conform to the following:

- The <saml:Issuer> element MUST be present and MUST contain the unique identifier of the requesting service provider. The Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- If the initial request was made over TLS and the <samlp:AuthnRequest> is to be signed, a <saml:Subject> element MAY be included in the request. If a <saml:NameID> is included to reference an existing user, subject information from the X.509 certificate SHOULD NOT be used for this purpose, as names used by the certificate authority may differ from those used by the providers.
- If the service provider wishes to permit the identity provider to establish a new identifier for the principal if none exists, it MUST include a <saml:NameIDPolicy> element with the AllowCreate attribute set to true.
- The <samlp:AuthnRequest> message MAY be signed (as directed by the SAML binding used).
2.5.2 <samlp:AuthnRequest> Message Processing Rules

If the identity provider cannot or will not satisfy the request, it MUST respond with a message containing an appropriate error status code or codes.

If the <samlp:AuthnRequest> is not authenticated and/or integrity protected, the information in it MUST NOT be trusted except as advisory. The <samlp:AuthnRequest> MUST be processed as follows:

- **Any** it is **RECOMMENDED** that any AssertionConsumerServiceURL or AssertionConsumerServiceIndex attributes in the <samlp:AuthnRequest> are **MUST be** verified as belonging to the entityID to whom the response will be sent.

- If the user agent cannot satisfy the <saml:SubjectConfirmation> present in the <samlp:AuthnRequest>, or it fails to obtain a key from the user agent, the identity provider MUST respond with a <samlp:Response> message containing an error status and no assertions.

- The identity provider is **NOT** obligated to honor the requested set of <saml:Conditions> in the <samlp:AuthnRequest>, if any.

2.5.3 <samlp:Response> Usage

If the identity provider wishes to return an error for this request, it **MUST NOT** include any assertions in the <samlp:Response> message. Otherwise, if the request is successful or the response is not associated with a request, the <samlp:Response> element **MUST** conform to the following:

- The <saml:Issuer> element of the <samlp:Response> **MAY be omitted**, but if present it **MUST** contain the unique identifier of the issuing identity provider; the Format attribute **MUST** be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

- The response **MUST** contain at least one <saml:Assertion>. Each assertion's <saml:Issuer> element **MUST** contain the unique identifier of the issuing identity provider, and the Format attribute **MUST** be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

- The set of one or more assertions **MUST** collectively contain one and only one <saml:AuthnStatement> that reflects the authentication of the principal to the identity provider.

- The assertion containing the <saml:AuthnStatement> is considered a holder-of-key assertion and **MUST** conform to [SAML2HoKAP]. Unless otherwise indicated by the service provider, the keying material **SHOULD** be contained in a <saml:SubjectConfirmation> element with at least one <saml:SubjectConfirmationData> element with a Method of urn:oasis:names:tc:SAML:2.0:cm:holder-of-key. Its <saml:SubjectConfirmationData> MUST embed the X.509 certificate directly in the <saml:SubjectConfirmationData> element by placing its base64 encoded representation in a <ds:KeyInfo> element.

- Additional <saml:SubjectConfirmation> elements **MAY** be included, though deployers should be aware of the implications of allowing weaker confirmation, as the processing is satisfy-
If the identity provider supports the Single Logout profile, defined in section 4.4 of [SAML2Prof], the `<saml:AuthnStatement>` MUST include a `SessionIndex` attribute or a uniquely identifying `<saml:NameID>` to enable per-session logout requests by the service provider.

Additional statements MAY be included in the assertion(s) at the discretion of the identity provider.

The assertion containing the `<saml:AuthnStatement>` MUST contain a `<saml:AudienceRestriction>` including the service provider's unique identifier as a `<saml:Audience>`.

Other conditions (and other `<saml:Audience>` elements) MAY be included as requested by the service provider or at the discretion of the identity provider. All such conditions MUST be understood by and accepted by the service provider in order for the assertion to be considered valid.

### 2.5.4 `<samlp:Response>` Message Processing Rules

Regardless of the SAML binding used, the service provider MUST do the following:

- Verify any signatures present on the assertion(s) or the response.

  - Verify that cryptographic data The `<saml:SubjectConfirmation>` MUST be confirmed in accordance with the requirements in [SAML2HoKAP] using the certificate resulting from successful mutual TLS authentication to the service provider by the user agent. strongly matches the X.509 certificate in the holder of key `<saml:SubjectConfirmationData>`. The service provider SHOULD NOT rely on any other data in the certificate to process the assertion but MAY utilize it more generally as additional information about the user.

- Verify that any assertions relied upon are valid according to processing rules in [SAML2Core].

Any assertion which is not valid, or whose subject confirmation requirements cannot be met, SHOULD be discarded and SHOULD NOT be used to establish a security context for the principal.

### 2.5.4.1 Artifact-Specific `<samlp:Response>` Message Processing Rules

If the HTTP Artifact binding is used to deliver the `<samlp:Response>`, the dereferencing of the artifact using the Artifact Resolution profile MUST be mutually authenticated, integrity protected, and confidential. Either the SAML binding used to dereference the artifact or message signatures can be used to authenticate the parties and protect the messages.

If the assertion is not encrypted, it is RECOMMENDED that the identity provider ensure that only the service provider to whom the `<samlp:Response>` message has been issued is given the message as the result of a `<samlp:ArtifactResolve>` request.

### 2.5.4.2 POST-Specific `<samlp:Response>` Message Processing Rules

If the HTTP POST binding is used to deliver the `<samlp:Response>`, the enclosed assertion(s) MUST be signed.
2.6 Use of Metadata

[SAML2Meta] defines endpoint elements to describe supported bindings and locations for providers. However, the metadata specification offers no way to distinguish the profile used by an endpoint. A boolean flag extension is not sufficient to signal use of this profile: because SAML implementations that don't implement this profile would ignore this optional attribute, they could send users to an inappropriate endpoint, potentially impacting interoperability and user experience. Rather than define new endpoint elements, it's sufficient to use the binding attribute to disambiguate betweenurn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key profile use andurn:oasis:names:tc:SAML:2.0:profiles:SSO:browser profile use. The actual binding to be used in this profile, as specified in [SAML2Bind], is instead placed into an extension attribute on the same endpoint. The combined information is sufficient to distinguish and utilize the correct profile and binding when making a request to an endpoint.

All <md:SingleSignOnService> endpoints and all <md:AssertionConsumerService> endpoints to be used exclusively with this profile MUST have a binding attribute of:


If an endpoint has the binding attribute

The following schema fragment defines the hok:Protocol attribute:

<attribute name="Protocol" type="anyURI"/>

This is an example <md:SingleSignOnService> endpoint to be used exclusively with this profile:

<SingleSignOnService
   Location="http://your-idp.example.org/some/path"/>

This is an example <md:AssertionConsumerService> endpoint to be used exclusively with this profile:

<AssertionConsumerService index="1" isDefault="true"
   Location="http://your-sp.example.org/some/path"/>

The <md:IDPSSODescriptor> element’s WantAuthnRequestsSigned attribute MAY be used by an identity provider to indicate a requirement that requests be signed. The <md:SPSSODescriptor> element’s AuthnRequestsSigned attribute MAY be used by a service provider to indicate the intention to sign all of its requests. If one of these attributes is present, the requirement SHOULD be met by counterparties.

The <md:SPSSODescriptor> element’s WantAssertionsSigned attribute MAY be used by a service provider to indicate a requirement that assertions delivered with this profile be signed. If present, the requirement SHOULD be met by counterparties.
3 Compatibility

This profile is derived from the Web Browser SSO Profile in [SAML2Prof]. The primary difference is the mandatory holder-of-key `<saml:SubjectConfirmation>` and the resulting mandate of client TLS authentication for user agent interactions. Because of its satisfy-any nature, inclusion of additional (in particular, bearer) `<saml:SubjectConfirmation>` must be done cautiously. An assertion including both holder-of-key and bearer subject confirmation could be issued in accordance with this profile and accepted as valid with no proof of possession of key, reintroducing attacks such as man-in-the-middle and replay.

The urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser(holder-of-key profile is technically compatible with the urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser profile, but it is RECOMMENDED that separate endpoints be used to ensure all processing is performed in accordance with each profile's requirements and avoid any negative impact on user experience.
4 Security and Privacy Considerations

Holder-of-key assertions, and protocols supporting their issuance and verification, in this profile have some different security and privacy characteristics from the bearer assertions used in the Web Browser SSO Profile.

- The identity provider's requirements for user authentication and keying material as described in section 2.4.4 can be simultaneously addressed by validation of an x.509 certificate presented by the user agent in TLS authentication from an issuer trusted by the identity provider, but this is not mandatory unless such an authentication context is requested by the service provider. If a certificate is used by the identity provider for principal authentication, pPPhishing is also eliminated, as there are greater challenges and no benefits to tricking the user into authenticating with legitimate credentials to a fraudulent party.

- There are limitations on the degree to which users can remain private under this profile since the X.509 certificate is presented to the service provider. Most end-user X.509 certificates have a unique distinguished name for the subject regularly containing personally identifying information. Additional information about the subject may be implicitly revealed through other fields. Furthermore, unless a new keypair is issued for every transaction, the public key is a de-facto persistent ID, as discussed in [SAML2Secure].

- Holder-of-key confirmation of the assertion issued eliminates the potential for assertion theft and encryption prevents privacy loss, eliminating man-in-the-middle attacks.

- The identity provider’s requirements for user authentication and keying material as described in section 2.4.4 can be simultaneously addressed by validation of an x.509 certificate presented by the user agent in TLS authentication from an issuer trusted by the identity provider, but this is not mandatory unless such an authentication context is requested by the service provider.

- Replay attacks that would have required tracking and checking assertion ID’s are prevented by validation of key possession.

- The <md:IDPSSODescriptor> element’s WantAuthnRequestsSigned attribute MAY be used by an identity provider to indicate a requirement that requests be signed. The <md:SPSSODescriptor> element’s AuthnRequestsSigned attribute MAY be used by a service provider to indicate the intention to sign all of its requests. If one of these attributes is present, the requirement SHOULD be met by counterparties. Deployers should consider the limited vulnerabilities associated with spoofed authentication requests and significant complexity resulting from authentication request signing.

- The session created by the service provider in the security context resulting from the Holder-of-Key Web Browser SSO Profile can be keyed by the TLS public key or session key. Application-layer sessions, such as maintained by cookies, are often poorly protected by user agents, allowing for theft of this session and impersonation of the user.
Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged. In addition, the editor would like to thank the National Institute of Informatics and the UPKI initiative for their support of this work.

Participants:

Scott Cantor, Internet2
Paul Friedrichs, Defense Information Services Agency
Patrick Harding, Ping Identity Corporation
Enrique de la Hoz, University of Alcalá de Henares
Toshiyuki Kataoka, National Institute of Informatics
Chad La Joie, SWITCH
Diego Lopez, RedIRIS
Tom Scavo, NCSA
David Waite, Ping Identity Corporation